Project #5, CUDA Monte Carlo
Vincent Rastello
rastellv@oregonstate.edu

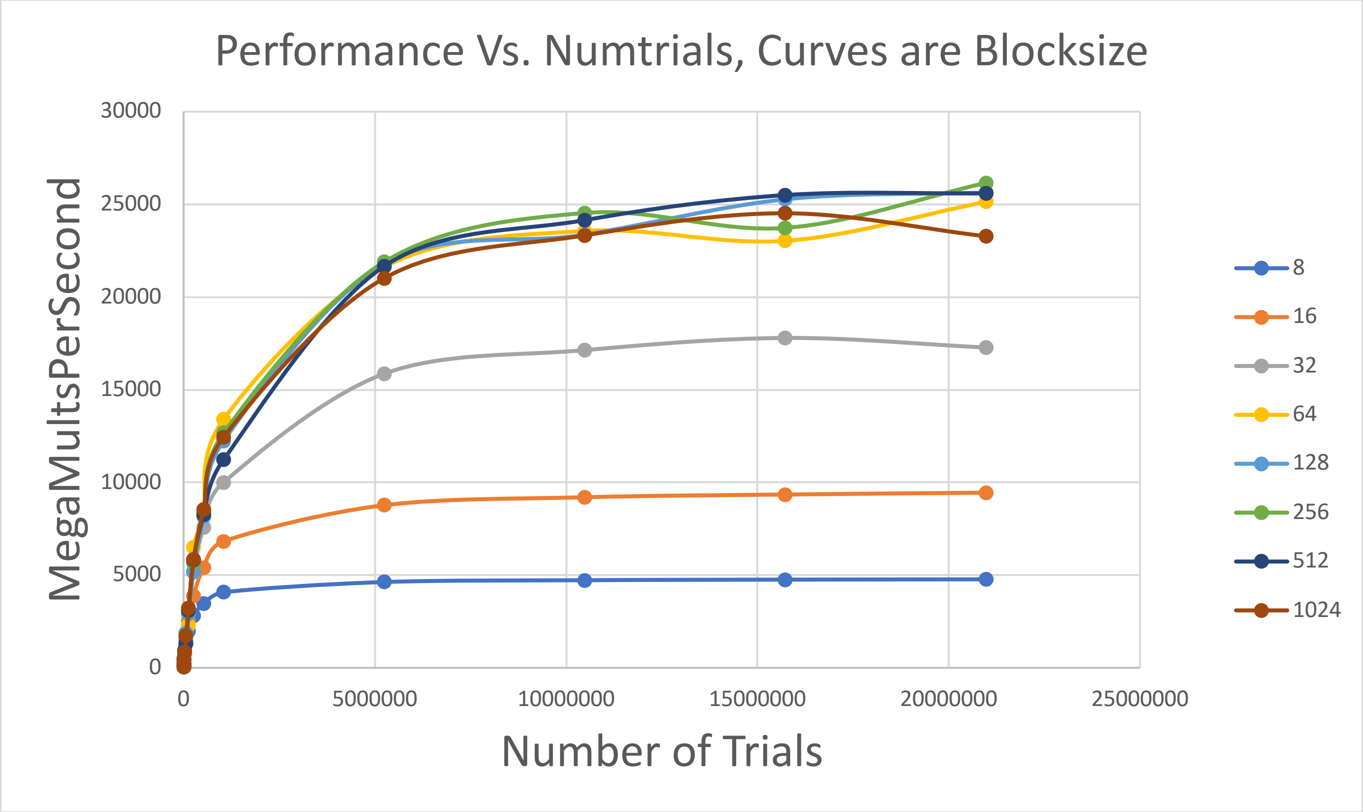
Project # 5

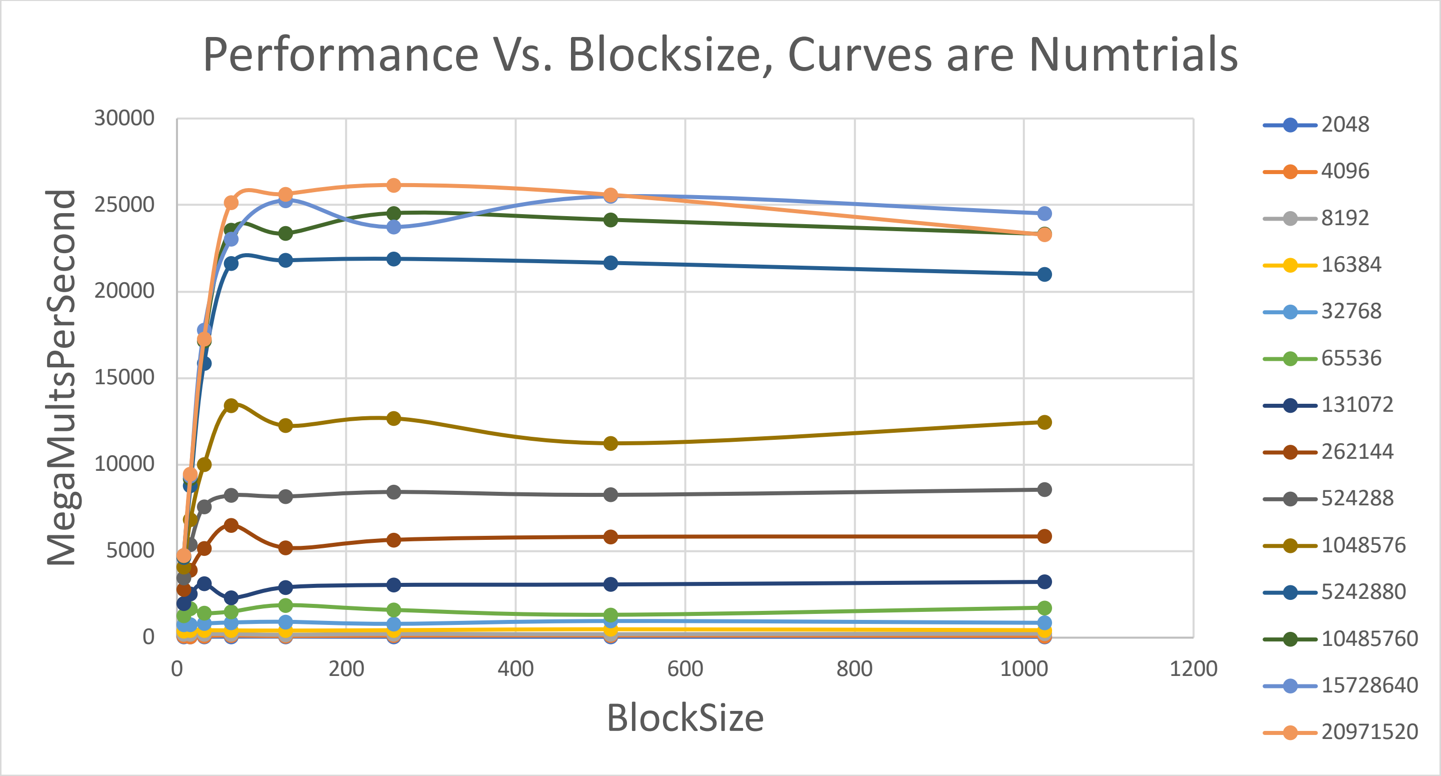
1. This was ran on the DGX-2 Server
   a. Each DGX server:
      i. Has 16 NVidia Tesla V100 GPUs
      ii. Has 28TB of disk, all SSD
      iii. Has two 24-core Intel Xeon 8168 Platinum 2.7GHz CPUs
      iv. Has 1.5TB of DDR4-2666 System Memory
      v. Runs the CentOS 7 Linux operating system

2. Table and Two Graphs

				Number of Threads per Block					
		8	16	32	64	128	256	512	1024
	2048	52.63	52.63	57.14	62.5	64.52	54.05	52.63	52.63
	4096	105.26	71.43	102.56	125	93.02	117.65	125	102.56
	8192	210.53	210.53	216.22	210.53	170.21	216.22	200	228.57
Number of trials	16384	400	421.05	432.43	421.05	410.26	432.43	484.85	432.43
	32768	759.64	764.18	818.55	877.46	917.56	800.63	962.41	858.34
	65536	1293.75	1682.83	1426.18	1508.1	1880.62	1602.5	1323.85	1732.66
	131072	1987.38	2525.28	3117.2	2312.82	2909.09	3047.62	3075.08	3222.66
	262144	2802.6	3899.1	5158.69	6496.43	5201.27	5653.55	5826.46	5847.25
	524288	3463.85	5394.8	7585.19	8224.9	8159.36	8419.32	8253.9	8551.15
	1048576	4075.12	6825.24	10008.55	13407.53	12249.72	12671.31	11233.46	12454.58
	5242880	4627.86	8771.35	15863.67	21623.33	21801.73	21892.04	21660.5	21013.21
	10485760	4725.77	9200.62	17135.39	23565.62	23363.99	24534.29	24149.16	23325.74
	15728640	4757.35	9348.22	17793.87	23032.8	25272.25	23730.03	25501.71	24519.61
	20971520	4773.58	9447.86	17279.51	25134.62	25642.07	26157.9	25593	23278.51

Project #5, CUDA Monte Carlo
Vincent Rastello
rastellv@oregonstate.edu





3. At 8 threads per block you see a flat curve holding at 5000 mega mults per second, 16 performance jumps up to 10,000, at 32 it jumps to 15000-18000. After 32, going to 64, 128, 256, 512 and 1024 blocksize all the curves even out around 25000 mega mults per second. Peak performance is reached for all blocksizes at roughly 10 million trials. As for performance vs blocksize, you see a pretty flat curvature after blocksize hits 64, with everything plateauing with increased block size. Trial count from 2K to 128K are all grouped in the below 5000 megamultspersecond, you see roughly a 2500 performance bump at each level above 128K until you reach 1M. I then upped the spacing by a factor of 5 to 5 million, that get's you up to above 20000 megamults persecond, all other values all the way up to 20 million trials doesn't exceed past 25000 megamults per second, which may be the ceiling for the DGX.
4. You are seeing low performance below 32 threads per block because each block has a minimum of 32 threads, anything below that and you have idling threads which ruins your performance. After 64 threads you don't see any performance improvement, this is because by doubling the thread count per block (32) you have maximized what you can parse out per block. After that increasing the number of threads per block has a negligible performance increase versus just using another block. It takes a fairly high number of trials 10 million to reach peak performance for each run. After 10 million trials there is no performance boost from increasing trial size. 10 million appears to be the largest value that can be broken up and still give a boost to performance.
5. Blocksize 16 and under is much worse because you are not using all the threads in the block, there are 32 threads per block by default. The idling threads amount to a drop in performance and efficiency.
6. The highest performance I saw in project 1 was 8 threads and 1 million trials, this topped out at 70 megamults per second. The DGX topped out at 25,000 megamults per second or 25 billion mults per second. This is because you are using literally 81,290 cores vs 8 cores. A problem with a 97% F parallel like the Monte Carlo run is going to take full advantage of that. From fully performing project 1 the DGX improved the Monte Carlo performance by a factor of 357x. If you were to take the Monte Carlo experiment and run it with one thread the megatrials per second came out to be roughly 9 in project 1. This means the speed up you achieve using the DGX is roughly 2800X.
7. This means that with the proper use of GPU parallel computing you can easily achieve speed ups of 2800X or more. With the right GPU set up you can easily transform large data and problems into easy problems.